REMARKS

Applicants appreciate the Examiner's thorough examination of the subject application and request reexamination and reconsideration of the subject application in view of the preceding amendments and following remarks. Applicants have carefully reviewed and considered the Office Action mailed on July 26, 2007, and the references cited therewith. Reconsideration and allowance of the subject application, as amended, are respectfully requested.

Claims 1-40 are pending in this application. As of this amendment, claims 1, 14, 25 and 33 have been amended. As a result, claims 1-40 are still pending in this application.

35 USC §102 Rejection of the Claims

Claims 25-40 were rejected under 35 USC § 102(b) as being anticipated by Douceur (U.S. Patent No. 6,067,547). Applicants respectfully traverse this rejection.

The Examiner admits on page 9 of the Official Action that Douceur does not "specifically teach finding a second index based on the information found at the location where the match was found, combining the second index with the original index to find the address where the data to be retrieved is stored." The Examiner relies upon Greene for teaching the proposition of finding a secondary index. Since claims 25 and 33 include limitations directed towards a "secondary array", Applicants respectfully submit that this rejection is moot. A more detailed discussion of the Greene reference with regard to the 35 USC 103 rejection is provided below.

35 USC §103 Rejection of the Claims

Claims 1-24 were rejected under 35 USC § 103(a) as being unpatentable over Douceur (U.S. Patent No. 6,067,547) and further in view of Greene (U.S. Patent No. 6,631,419). Applicants respectfully traverse this rejection.

In an effort to advance prosecution, Applicants have amended independent claim 1, which is provided below for the Examiner's convenience.

> 1. (Currently Amended) Apparatus for use in data retrieval, comprising:
> a hash value generator configured to generate an index value and a signature value, based on input data,

> a first memory array configured to receive the index value as an address, said first memory array is further configured to output, in response to said index value, a data unit comprising a plurality of signature values arranged in respective positions in said data unit~~,~~ ; and
>
> a comparator configured to receive the signature value and said data unit from the first memory array, the comparator is further configured to compare the signature value with the plurality of signature values of said data unit, and if a match is found, said comparator is further configured to generate an address signal including said address of said index value and the position corresponding to the matching signature values~~, and~~ in a second memory array, said ~~a~~ second memory array configured to receive said address signal including ~~of~~ said index value and the position corresponding to the matching signature values and, in response thereto, output payload data without serially searching a linked list associated with said second memory array. (Emphasis Added).

Thus, Applicants have amended claim 1 in an effort to advance prosecution and to more clearly define the avoidance of searching a linked list in a serial manner. The difficulties inherent in serially searching a linked list are discussed throughout the subject application. For example, in paragraphs [0015-0016] of the subject application, which have been provided below for the Examiner's convenience.

> [0015] The index value 112 and the position value 142 may be input to the secondary data array 130 merged as an address signal. In response, the secondary data array 130 may output a unit of payload data 132 from within the array 130. The payload data can be returned in response to the lookup command.
>
> [0016] The foregoing embodiment avoids the latencies inherent in linked list hashing systems noted above. Regardless of the rate of collisions among index values, payload data can be retrieved from a memory system with only two memory reads. Thus, while the given index value still may not represent the input data uniquely, the foregoing embodiments resolve ambiguities through use of a signature in the primary data array. The signature's position helps determine the payload data's position in memory and permits the system to avoid traipsing across multiple entries as in the linked list system.

In contrast, Applicants respectfully submit that Douceur teaches away from the Applicants' current invention by teaching precisely what the present application is attempting to avoid. Namely, the searching across multiple entries in a linked list as described above. The Examiner points to column 3, line 50 to column 4 line 45 of Douceur as describing "how a hash of input data is generated from input data, and how the hash is used to locate a pointer to where the data record corresponding to the input signal is identified by searching for a match with a key." Official Action page 8. The aforementioned section of Douceur is provided below for the Examiner's convenience.

> These two steps are together referred to as a hash function, and the result is called a hash value. In the third step, the hash value is reduced to an address in the valid range by taking the value modulo the size of the table. If the table size is a power of two, this is equivalent to taking the least significant bits of the hash value. For example, if the table size is 8, the 3 (log.sub.2 8) least significant bits of the hash value can be used to form the address.
>
> Address generation must be performed for both insertion and retrieval operations. For retrieval, the next step is to scan through the linked list pointed to by the addressed location in the hash table. The key in each record in the list must be examined to determine whether it matches the search key. If no records have matching keys, then the search is unsuccessful.
>
> On average, each successful search will examine a number of records equal to half of the mean list length, but each unsuccessful search will examine a number of records equal to the entire mean list length. If unsuccessful searches are expected to be common, their performance can be improved by storing the records in each list in sorted order. If the records in each list are sorted by their keys, a search through the list can terminate as soon as a record is found whose key value is greater than the search key value, since if a record containing the search key were present, it would occur in the list prior to the record with the greater key value.
>
> For example, note that the records in FIG. 1 are organized in linked lists, in sorted order. Consider again the example search discussed above for a record with a key value of "D". The address generation function produces an address of 1 for this key. Entry 1 points to a record having key value "P", which is greater than the value of the search key. If a record with key value "D" were present in the list, it would precede the record with key value "P". Thus, it can be concluded that no record with key value "D" is stored in the hash table--the search can be terminated without scanning through the entire linked list.
>
> A further performance improvement can be made by storing a signature value of the key along with each record. One way to calculate this signature is to use some bits of the hash value calculated for the key. Obviously, these bits must be at different positions than those for the address itself, since all records in the same linked list have the same address. When the search is performed, the signature values are compared rather than the entire search key. Since the signature values are often significantly smaller than the key values, comparison times can be reduced by this technique. Only if the signature values match do the actual key values need to be compared. If signature values are employed, then the records in each linked list can be sorted according to the signatures instead of the key values. This optimization can improve the search times for both unsuccessful and successful searches. Douceur, col. 3, line 50- col. 4, line 45. (Emphasis Added).

Thus, Douceur appears to teach the use of signature values to sort records in a linked list. "This optimization can improve the search times for both unsuccessful and successful searches." Thus, as shown above, Applicants respectfully submit that Douceur teaches searching through a linked list. In fact, the passages cited by the Examiner discuss the optimization of this searching. Again, Applicants respectfully submit that searching through a linked list in this manner teaches away from the Applicants' present invention. Douceur appears to teach the optimization of a

search, which may reduce the amount of searching necessary, however, the searching of a linked list is still required. Applicants' newly amended claim 1 does not require this searching.

As such, Applicants respectfully submit that neither Douceur or Greene teach a comparator being "further configured to generate an address signal including said address of said index value and the position corresponding to the matching signature values in a second memory array, said second memory array configured to receive said address signal including said index value and the position corresponding to the matching signature values and, in response thereto, output payload data without serially searching a linked list associated with said second memory array."

Therefore, Applicants respectfully submit that neither Douceur nor Greene, either alone or in combination teach each and every limitation of Applicants' claim 1. It is Applicants' understanding that Greene merely discloses the concept of a second memory array and does not reconcile the deficiencies created by Douceur (i.e., the searching of a linked list).

Thus, Applicants respectfully submit that newly amended claim 1 is in condition for allowance. Independent claims 14, 25, and 33 have been amended to include similar limitations and are believed to be in condition for allowance as well. Since claims 2-13, 15-24, 26-32, and 34-40 depend, either directly or indirectly, from Applicants' independent claims, Applicants submit that these claims are in condition for allowance as well.

Applicant respectfully submits that the claims are in condition for allowance and notification to that effect is earnestly requested. The Examiner is invited to telephone Applicant's attorney (603-668-6560) to facilitate prosecution of this application.

If necessary, please charge any additional fees or credit overpayment to Deposit Account No. 50-2121.

Respectfully submitted,

JON H. KRUEGER ET AL.

By their Representatives,

CUSTOMER NUMBER: 45459

Telephone Number: 603-668-6560

By / Edmund P. Pfleger /
Edmund P. Pfleger
Reg. No. 41,252